Refine Search

Search Results -

Term	Documents
"7191196"	2
7191196S	0
"7191196".PN..PGPB,USPT,USOC,EPAB,JPAB,DWPI,TDBD.	2
(7191196.PN.).PGPB,USPT,USOC,EPAB,JPAB,DWPI,TDBD.	2

Database:
US Pre-Grant Publication Full-Text Database
US Patents Full-Text Database
US OCR Full-Text Database
EPO Abstracts Database
JPO Abstracts Database
Derwent World Patents Index
IBM Technical Disclosure Bulletins

Search: L11

Refine Search | Recall Text | Clear | Interrupt

Search History

DATE: Monday, July 23, 2007 **Purge Queries** Printable Copy Create Case

Set Name side by side	Query	Hit Count	Set Name result set
DB=PGPB,USPT,USOC,EPAB,JPAB,DWPI,TDBD; PLUR=YES; OP=OR			
L11	7191196.pn.	2	L11
DB=USPT; PLUR=YES; OP=OR			
L10	l4 and multidimen$	1	L10
L9	L8 and construct$	1	L9
L8	L6 and populat$	1	L8
L7	L6 and inflat$	0	L7
L6	L5 and instant$	1	L6
L5	7010523.pn.	1	L5
L4	L3 and inflat$	1	L4
L3	L1 and instantiat$	1	L3
L2	L1 and intantiat$	0	L2

L4: Entry 1 of 1 File: USPT Mar 13, 2007

DOCUMENT-IDENTIFIER: US 7191196 B2
TITLE: Method and system for maintaining forward and backward compatibility in flattened object streams

PRIOR-PUBLICATION:

DOC-ID	DATE
US 20030005169 A1	January 2, 2003

Brief Summary Text (6):
Using object serialization, an object can be persisted and then reused. Object serialization is the process of saving an object's state to a sequence of bytes as well as the process of rebuilding those bytes into a live object, although the latter is sometimes called "deserialization". The complete state of an object can be written to an output stream, and at some later time, the object can be recreated by reading its serialized state from an input stream. Using a three-dimensional analogy for an object, a serialized object is often called a "flattened" object, while a recreated object is sometimes called an "inflated" object.

Brief Summary Text (16):
At some subsequent point in time, the data stream representing a serialized object may be read. An object will be instantiated in accordance with the class identifier from the data stream, at which point the class version number of the instantiated object and the class version number of the serialized object may differ. Serialized attribute values from the data stream are stored within the instantiated object in accordance with the class version number and attribute version number relationships between the serialized object and the instantiated object.

Description Paragraph (27):
FIG. 2B shows object 210, which is structured in accordance with Version 1 of the Person class. Object 210 undergoes serialization operation 212 to create flattened object data stream 214. Data stream 214 then undergoes deserialization operation 216 to inflate object 218, which can be seen to be a copy of the original object 210.

Description Paragraph (29):
FIG. 2C shows object 220, which is structured in accordance with Version 2 of the Person class. Object 220 undergoes serialization operation 222 to create flattened object data stream 224. Data stream 224 then undergoes deserialization operation 226 to inflate object 228, which can be seen to be a copy of the original object 220.

Description Paragraph (30):
The simplest and most frequent scenario for serializing an object occurs when the version of the class that is read out of the stream is the same as the version that was written. This scenario, as it would appear using the standard serialization mechanism, is represented in FIGS. 2B 2C. As the figures show, when the object is serialized, information that identifies the object's class and the value of the

Hit List

First Hit | Clear | Generate Collection | Print | Fwd Refs | Bkwd Refs | Generate OACS

Search Results - Record(s) 1 through 2 of 2 returned.

☐ 1. Document ID: US 7191196 B2

L11: Entry 1 of 2 File: USPT Mar 13, 2007

US-PAT-NO: 7191196
DOCUMENT-IDENTIFIER: US 7191196 B2

TITLE: Method and system for maintaining forward and backward compatibility in flattened object streams

DATE-ISSUED: March 13, 2007

PRIOR-PUBLICATION:

DOC-ID	DATE
US 20030005169 A1	January 2, 2003

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Perks; Michael Albert	Austin	TX		US
Sundberg; Sean Michael	Waco	TX		US

ASSIGNEE-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY	TYPE CODE
International Business Machines Corporation	Armonk	NY		US	02

APPL-NO: 09/894096 [PALM]
DATE FILED: June 28, 2001

INT-CL-ISSUED:

TYPE	IPC	DATE	IPC-OLD
IPCP	G06F17/30	20060101	G06F017/30

INT-CL-CURRENT:

TYPE	IPC	DATE
CIPP	G06 F 17/30	20060101

US-CL-ISSUED: 707/203; 707/202
US-CL-CURRENT: 707/203; 707/202

FIELD-OF-CLASSIFICATION-SEARCH: 707/6, 707/103R, 707/103Y, 707/203, 707/202, 345/619, 717/108, 719/330

See application file for complete search history.

PRIOR-ART-DISCLOSED:

U.S. PATENT DOCUMENTS

PAT-NO	ISSUE-DATE	PATENTEE-NAME	US-CL
5499365	March 1996	Anderson et al.	707/203
6298353	October 2001	Apte	707/103R
6301585	October 2001	Milne	707/103R
6477701	November 2002	Heistermann et al.	717/108

ART-UNIT: 2165

PRIMARY-EXAMINER: Mofiz; Apu M.

ATTY-AGENT-FIRM: Burwell; Joseph R. Rodriguez; Herman Van Leeuwen; Leslie

ABSTRACT:

A method, a system, an apparatus, and a computer program product are presented for object-oriented management of serializable objects. An object has a set of attributes and a class version number, which identifies the object as an instance of a specific version of a class. In addition, each attribute in the object has an attribute version number, which may represent a version of a class in which the associated attribute was initially declared within the class. The object may be serialized into a data stream, and the data stream has the class version number of the object, an attribute value for each attribute in the set of attributes, and an attribute version number for each attribute in the set of attributes. The data stream may also have a class identifier for the class of the object and an attribute count for the set of attributes into the data stream.

11 Claims, 18 Drawing figures

---

2. Document ID: US 7191196 B2, US 20030005169 A1

L11: Entry 2 of 2 File: DWPI Mar 13, 2007

DERWENT-ACC-NO: 2003-352942
DERWENT-WEEK: 200721

TITLE: Version management of serialized object streams in object oriented applications, involves associating class version identifier with object and attribute version identifier with attribute

INVENTOR: PERKS, M A; SUNDBERG, S M

PRIORITY-DATA: 2001US-0894096 (June 28, 2001)

PATENT-FAMILY:

PUB-NO	PUB-DATE	LANGUAGE	PAGES	MAIN-IPC
US 7191196 B2	March 13, 2007		000	G06F017/30
US 20030005169 A1	January 2, 2003		017	G06F009/44

INT-CL (IPC): G06F 9/44; G06F 17/30

ABSTRACTED-PUB-NO: US20030005169A
BASIC-ABSTRACT:

NOVELTY - An object comprising a set of attributes is identified. A class version identifier associated with the object identifies the object as an instance of a specific version of a class. An attribute version identifier is associated with an attribute in the set of attributes.

DETAILED DESCRIPTION - INDEPENDENT CLAIMS are included for the following:

(1) method for providing backwards and forwards compatibility between versions of serialized object data;

(2) computer program product for object oriented management of serialized objects;

(3) computer program product for providing backward and forwards compatibility between different versions of serialized object data;

(4) apparatus for object oriented management of serializable objects; and

(5) apparatus for providing backwards and forwards compatibility between versions of serialized object data.

USE - For object oriented management of serializable objects.

ADVANTAGE - Facilitates object serialization and de-serialization, thereby maintaining forward and backward compatibility for flattened objects across different versions of a class, hence prevents error when version mismatches are detected between flattened and inflated data objects.

DESCRIPTION OF DRAWING(S) - The figure shows an object oriented methodology supporting serialization and de-serialization with version management.

Term	Documents
"7191196"	2
7191196S	0
"7191196".PN..PGPB,USPT,USOC,EPAB,JPAB,DWPI,TDBD.	2
(7191196.PN.).PGPB,USPT,USOC,EPAB,JPAB,DWPI,TDBD.	2

Display Format: - Change Format

Previous Page Next Page Go to Doc#

L9: Entry 1 of 1 File: USPT Mar 7, 2006

DOCUMENT-IDENTIFIER: US 7010523 B2
TITLE: System and method for online analytical processing

PRIOR-PUBLICATION:

DOC-ID	DATE
US 20040133552 A1	July 8, 2004

Brief Summary Text (15):
One aspect of the invention involves the specification and construction of queries in the OLAP application in an object-oriented representation rather than by a textual query such as SQL. The query objects resemble the business model of the OLAP application rather than the relational database model of the data warehouse. The execution of the query objects can be performed by a separate data server for providing the OLAP services, with the OLAP application holding remote references to the query objects. More specifically, the objects represent the query state, and the OLAP application refines the queries by invoking methods on the query objects. These actions cause corresponding methods to be invoked on the data server objects remotely. In this configuration, the data server can easily determine how the query has been altered and thus perform any conversions necessary between the object representation and the relational database's own query representation without encumbering the OLAP application itself.

Description Paragraph (27):
Accordingly, metadata is used to inform the OLAP application 101 about the data that is available within the relational database 121 in a manner so that the OLAP application 101 can define multidimensional objects for analysis. When the OLAP application 101 runs, the OLAP application 101 instantiates these multidimensional objects and populates them with data fetched from the database.

Description Paragraph (30):
Dimensions identify and categorize the OLAP application's data. In a relational database system, dimension members are stored in a dimension table. Each column represents a particular level in a hierarchy. In a star schema, the columns are all in the same table; in a snowflake schema, the columns are in separate tables for each level. Because measures are typically multidimensional, a single value in a measure must be qualified by a member of each dimension to be meaningful. For example, a Sales measure might have dimensions for Product, Geographic Area, and Time. A value in the Sales measure (37854) is only meaningful when it is qualified by a product (DVD Player), a geographic area (Pacific Rim), and Time (March 2001). Defining a dimension in the data warehouse creates a database dimension object, in addition to creating metadata. A dimension object contains the details of the parent-child relationship between columns in a dimension table; it does not contain data. The database dimension object is used by the Summary Advisor and query rewrite to optimize the data warehouse. However, on the multidimensional side, a dimension does contain data, such as the names of individual products, geographic areas, and time periods. The OLAP API uses the metadata, dimension objects, and dimension tables to construct its dimensions.

Description Paragraph (43):
FIG. 2 is a flowchart illustrating the life cycle of an OLAP query in accordance with one embodiment of the present invention. The OLAP application 101 begins by connecting to a metadata provider (step 201) and viewing the metadata (step 203). As described in more detail hereinafter, a metadata provider is responsible for obtaining the metadata from the relational database 121. If the client running the OLAP application 101 wishes to view the real data (tested at step 205), then the OLAP application 101, in response, connects to another object called a data provider (step 207). Upon connection, the client creates objects called sources, which are used by the OLAP API 103 to construct and represent queries. Typically, the client creates many sources, beginning with one that represents the basic metadata objects that are found in the relational database 121 (step 209). For example, the client might create a query by specifying a subset of the values in a basic measure.

CLAIMS:

1. A computer-implemented method for analyzing data comprising the steps of: receiving a call from an analytical processing application; constructing a query object based on the call; translating the query object into a textual query for submission to a data warehouse; retrieving data from the data warehouse in response to the submission of the textual query; and providing at least some of the data retrieved from the data warehouse in response to the textual query; wherein said providing at least some of the data retrieved includes: receiving a specification of an extent of a multidimensional cursor from the analytical processing application; and determining said at least some of the data based on the specification.

6. The method of claim 1, wherein said constructing the query object based on the call includes: generating a source object; and generating a cursor object, wherein the source object includes a specification for the query, wherein the cursor object includes a result of the query.

Previous Doc Next Doc Go to Doc#